WHAT WE CLAIM IS:

1. A processor having:

respective first and second external instruction formats in which instructions are received by the processor, each instruction having an opcode which specifies an operation to be executed, and each external format having one or more preselected opcode bits in which the opcode appears;

an internal instruction format into which instructions in the external formats are translated prior to execution of the operations;

wherein:

the operations comprise a first operation specifiable in both said first and second external formats, and a second operation specifiable in said second external format;

said first and second operations have distinct opcodes in said second external format; and

in each said preselected opcode bit which the first and second external formats have in common, the opcodes of the first operation in the two external formats are identical.

2. A processor as claimed in claim 1, wherein:

the operations comprise one or more further first operations, each specifiable in both said first and second external formats, and one or more further second operations specifiable in said second external format;

for every pair of operations, made up of one said first operation and one said second operation, the operations of the pair have distinct opcodes in said second external format; and

in each said preselected opcode bit which the first and second external formats have in common, the opcodes of each first operation in the two external formats are identical.

3. A processor as claimed in claim 1, having:

a third external instruction format in which instructions are received by the processor, each instruction having an opcode which specifies an operation to be executed, and said third external format having one or more preselected opcode bits in which the opcode appears;

respective first and second internal instruction formats into which instructions in the external formats are translated prior to execution of the operations;

wherein:

said second operation is specifiable in both said second and third external formats;

an instruction specifying said first operation in either said first or second external format is translated into said first internal format, and an instruction specifying said second operation in either said second or third external format is translated into said second internal format; and

in each said preselected opcode bit which the second and third external formats have in common, the opcodes of the second operation in the two external formats are identical.

4. A processor as claimed in claim 3, wherein:

the operations comprise one or more further first operations, each specifiable in both said first and second external formats, and one or more further second operations specifiable in said second external format;

for every pair of operations, made up of one said first operation and one said second operation, the operations of the pair have distinct opcodes in said second external format;

in each said preselected opcode bit which the first and second external formats have in common, the opcodes of each first operation in the two external formats are identical; and in each said preselected opcode bit which the second and third external formats have in common, the opcodes of each first operation in the two external formats are identical.

5. A processor as claimed in claim 1, being a VLIW processor, wherein one external format is a scalar instruction format used for scalar instructions, and another external format is a VLIW instruction format used for VLIW instructions.

6. A processor as claimed in claim 1, being a VLIW processor, wherein the external formats are or comprise two different VLIW formats.

7. A processor as claimed in claim 6, wherein the two different VLIW formats are used in different respective instruction slots of a VLIW instruction parcel.

8. A processor as claimed in claim 6, wherein at least one instruction slot of a VLIW instruction parcel uses the two different VLIW formats.

9. A processor as claimed in claim 1, wherein one external format has an instruction width different from that of another external format.

10. A processor as claimed in claim 1, having:

a translation unit which performs a predetermined translation operation to translate each said external-format opcode into a corresponding internal-format opcode.

11. A processor as claimed in claim 10, wherein said translation operation involves selecting and/or permuting bits amongst said preselected opcode bits in the external-format instruction.

12. A processor as claimed in claim 10, wherein the translation operation is independent of the external-format opcode.

13. A processor as claimed in claim 12, wherein the translation unit identifies the internal format

into which each external-format instruction is to be translated, and carries out said translation operation according to the identified internal format.

14. Processor instruction encodings having:

respective first and second external instruction formats in which the instructions are received by a processor, each instruction having an opcode which specifies an operation to be executed, and each external format having one or more preselected opcode bits in which the opcode appears;

an internal instruction format into which the processor instructions in the external formats are translated prior to execution of the operations;

wherein:

a first operation executable by the processor is specifiable in both said first and second external formats, and a second operation executable by the processor is specifiable in said second external format;

said first and second operations have distinct opcodes in said second external format; and

in each said preselected opcode bit which the first and second external formats have in common, the opcodes of the first operation in the two external formats are identical.

15. A method of encoding processor instructions for a processor having respective first and second external instruction formats in which instructions are received by the processor, each instruction having an opcode which specifies an operation to be executed, and each external format having one or more preselected opcode bits in which the opcode appears, the processor also having an internal instruction format into which instructions in the external formats are translated prior to execution of the operations, and the operations comprise a first operation specifiable in

both said first and second external formats, and a second operation specifiable in said second external format, said method comprising:

encoding said first and second operations with distinct opcodes in said second external format; and

encoding the opcodes of the first operation in said first and second external formats so that, in each said preselected opcode bit which the first and second external formats have in common, the opcodes of the first operation in the two external formats are identical.

16. A method of encoding instructions for a processor having two or more external instruction formats and one or more internal instruction formats, the method comprising:

(a) selecting initial encoding parameters including a number of effective opcode bits in each external and internal format and a set of mapping functions, each said mapping function serving to translate an opcode specified by said opcode bits in one of the external formats to an opcode specified by said opcode bits in the, or in one of the, internal formats;

(b) allocating each operation executable by the processor an opcode distinct from that allocated to each other operation in each external and internal format in which the operation is specifiable, the allocated opcodes being such that each relevant mapping function translates such an external-format opcode allocated to the operation into such an internal-format opcode allocated to the operation and such that all the internal-format opcodes allocated to the operation have the same effective opcode bits; and

(c) if in the allocation (b) no opcode is available for allocation in each specifiable format for every one of said operations, determining which of said encoding parameters is constraining the allocation (b), relaxing the constraining parameter, and then repeating the allocation (b).

17. A method as claimed in claim 16, wherein each said mapping function involves selecting all bits of the external-format opcode as some or all of the bits of the internal-format opcode.

18. A method as claimed in claim 16, wherein in the selection (a), for each external and internal format, said number of effective opcode bits is made equal to a minimum possible number of opcode bits that could theoretically encode the number of operations specifiable in the format concerned.

19. A method as claimed in claim 16, wherein the allocation (b) comprises a series of iterations, and prior to commencing the series of iterations a set of available opcodes in each external and internal format is formed, and in each iteration of the series one said operation is considered and the allocation of the opcode to the considered operation is made based on an examination of the sets of available opcodes in each external and internal format in which the considered operation is specifiable.

20. A method as claimed in claim 19, wherein, for each said external and internal format, the set of available opcodes formed prior to commencing a series of iterations has a number of members dependent upon said number of effective opcode bits currently applicable to that format.

21. A method as claimed in claim 19, wherein the available opcodes in all the sets have the same working number of bits.

22. A method as claimed in claim 21, wherein said working number is set equal to a minimum possible number of opcode bits that could theoretically encode the number of operations specifiable in the external or

internal format having the highest number of operations specifiable in the format concerned.

23. A method as claimed in claim 19, wherein each said iteration of the allocation (b) comprises:

(b-1) determining which, if any, available opcodes are common to the sets for all the external and internal formats in which the considered operation is specifiable; and

(b-2) if the determination in (b-1) is that one or more such available opcodes are common, selecting the or one of the common opcodes, allocating it to the considered operation, and removing the selected opcode from the set for each external and internal format in which the considered operation is specifiable.

24. A method as claimed in claim 23 wherein each said iteration of the allocation (b) further comprises:

(b-3) if the determination in (b-1) is that no common available opcode is present in the sets for all the external and internal formats in which the considered operation is specifiable, making all existing allocated opcodes void and carrying out the determination and relaxation (c).

25. A method as claimed in claim 16, further comprising:

(d) after all of the operations have been allocated one of said available opcodes having said working number of bits, determining for each external format whether that working number is greater than a minimum number of bits needed to provide each operation specifiable in that external format with its own distinct opcode and, if so, restricting the allocated opcodes in that external format to the determined minimum number of bits.

26. A method as claimed in claim 25, wherein the operations in (d) comprise:

(d-1) identifying for each external format a maximum-length common prefix, if any, for all allocated opcodes in the external format concerned; and

(d-2) removing the identified common prefix from all the allocated opcodes in the external format concerned; and

(d-3) adjusting each mapping function that serves to translate an opcode specified by the opcode bits in the external format concerned into an opcode specified by internal-format opcode bits so that the mapping function prepends the identified common prefix to the external-format opcode bits during translation.

27. A method as claimed in claim 16, wherein if the determination in (c) is that the number of effective opcode bits in one of the external or internal formats is the constraining parameter, the number of effective opcode bits in that format is increased.

28. A method as claimed in claim 16, carried out by an electronic data processing device.

29. A computer-readable recording medium storing a program which, when executed, encodes instructions for a processor having two or more external instruction formats and one or more internal instruction formats, said program comprising:

(a) a selecting code portion which selects initial encoding parameters including a number of effective opcode bits in each external and internal format and a set of mapping functions, each said mapping function serving to translate an opcode specified by said opcode bits in one of the external formats to an opcode specified by said opcode bits in the, or in one of the, internal formats;

(b) an allocating code portion which allocates each operation executable by the processor an opcode distinct from that allocated to each other operation in each external and internal format in which the operation is specifiable, the allocated opcodes being such that each relevant mapping function translates such an external-format opcode allocated to the operation into such an internal-format opcode allocated to the operation and such that all the internal-format opcodes allocated to the operation have the same effective opcode bits; and

(c) a determining code portion which, if no opcode is available to the allocating code portion for allocation in each specifiable format for every one of said operations, determines which of said encoding parameters is constraining the allocation in step (b), and relaxes the constraining parameter, and then causes the allocating code portion to repeat the allocation of opcodes.

09502208.021401